Maybe you saw my post about Bloc Jams already, but if you haven’t and you’re curious, you can see it <here>. After completing the Bloc Jams project in JavaScript and jQuery, I reconfigured the project into AngularJS. The product features remained the same, but the code base wound up looking very different.

With the Angular conversion, my 270-line `album.js` script from the original Bloc Jams project dropped down to about 190 lines of code. For example, the functions for skipping to the next or previous song in the JavaScript/jQuery version took 41 lines of code. In Angular I did it in about 26 lines (not including declarations).

< jquery photo >

<angular photo>

When learning Angular, I struggled a bit with understanding data bindings. Unlike JavaScript and jQuery which ties scripts to a general HTML page, in Angular, to make sure the correct data from the script displays correctly in the UI, you must bind the data within the HTML. For me, this meant that I had to put more thought into HTML formatting to make sure that the UI would display the necessary information.

<jquery photo>

<angular photo>

Also, rather than having one long code base for each HTML page, Angular breaks the code up into controllers, services, filters and directives so that different sections of code can be used independently. Although this makes for good architecture, the abstraction of thought required to break your code down this way can be tough! It also results in more script files: in Angular I had nine files while in JavaScript/jQuery I only had five.

By refactoring a project and code base I was familiar with, this approach to my introduction to Angular allowed me to pick up on the concepts quickly!